From Brouwer's Thesis to the Fan Functional

Ulrich Berger
Swansea University

University of Birmingham
Theoretical Computer Science Seminar
February 21, 2020

Overview

1. Introduction
2. Brouwer's thesis
3. Abstract bar induction
4. Vacuous truth
5. Proving uniform continuity
6. Extracting the fan functional

Introduction

What are the logical roots of intriguing algorithms/computing principles?

- Primitive recursion comes from induction on **N**.
- General recursion comes from wellfounded induction.
- The extended Euklidean algorithm comes from a classical proof that **Z** is a principal ideal ring.
- Normalization by evaluation (for the typed lambda-calculus) comes from the Tait/Girard proof of strong normalization, respectively a completeness proof for intuitionistic logic.
- ...

Where does Tait's fan functional come from?

The fan functional

The fan functional computes for every continuous function on Cantor space with values in $\mathbf{N}$ its least modulus of uniform continuity:

$$\mathbf{FAN} : (\{0,1\}^{\mathbf{N}} \to \mathbf{N}) \to \mathbf{N}$$
$$\mathbf{FAN}(F) = \mu n \forall \alpha, \beta\, (\alpha =_n \beta \to F\,\alpha = F\,\beta)$$

where $\alpha =_n \beta \stackrel{\text{Def}}{=} \forall k \in \mathbf{N}\,(k < n \to \alpha\, k = \beta\, k)$.

So, clearly, this must come from:

Fan theorem:

Every continuous function on Cantor space with values in $\mathbf{N}$ is uniformly continuous.

The real question is what are the right logical and mathematical principles and what is the right formal system for a proof of this theorem in order to extract the fan functional, more precisely, a purely functional program that computes it?

Brief history of the fan functional

Tait introduced the Fan functional in 1963 and showed that it is recursively continuous but not computable by Kleene's schemata S1-S9, thus shattering Kleene's hope that S1-S9 is a universal notion of computation in higher types.

Tait's proof of non-S1-S9-computability had a gap which was noted and fixed in various ways by Gandy, Hyland, Normann, Escardo ...

The above holds if S1-S9 is interpreted in the hereditarily *total* continuous functionals by Kleene and Kreisel (1959). If, however, S1-S9, is interpreted in the *partial* continuous functionals by Scott and Ershov (1970s), then there is an S1-S9 (equivalently PCF) definable functional whose restriction to total arguments is the Fan functional (Gandy 19??, B 1990).

In 2000, Normann proved that, in fact, *every* total recursive functional can be S1-S9 defined in this way, so S1-S9 *is* universal, when interpreted in the 'right' way.

In 2015, Escardo and Oliva generalized the Fan functional to the notion of 'products of selection functionals' and applied it to game theory.

Brouwer's thesis

Brouwer's thesis (**BT**) Every bar is inductive.

A predicate P on natural numbers is a *bar* if $\forall\alpha\exists n\, P(\bar{\alpha}\, n)$

P is an *inductive bar* if $\mathbf{IB}_P(\langle\rangle)$ holds where, inductively,

(i) If $P(s)$, then $\mathbf{IB}_P(s)$.

(ii) If $\mathbf{IB}_P(s * n)$ for all $n \in \mathbf{N}$, then $\mathbf{IB}_P(s)$.

More compactly,

$$\mathbf{IB}_P(s) \stackrel{\mu}{=} P(s) \vee \forall n\, \mathbf{IB}_P(s * n) \qquad (\mu \text{ means 'least'})$$

Hence **BT** can be written as the schema

$$\forall\alpha\exists n\, P(\bar{\alpha}\, n) \to \mathbf{IB}_P(\langle\rangle)$$

Bar induction for decidable bars (**BI**)

If

(1) P is a bar,

(2) P decidable and $P \subseteq Q$,

(3) $\forall s\,(\forall n\, Q(s * n) \to Q(s))$,

then $Q(\langle\rangle)$.

Where 'P decidable' means $\forall n\,(P(n) \vee \neg P(n))$.

It is easy to see that **BT** implies **BI**.

Issues with **BT** and **BI** (regarding applicability)

BT $\forall\alpha\exists n\, P(\bar{\alpha}\, n) \to \mathbf{IB}_P(\langle\rangle)$

- restricted to natural numbers
- talks about infinite sequences
- the premise has computational content which is often not available
- the conclusion has unwanted computational content
- decidability of the bar P (required in **BI**) is too restrictive

Therefore, we *weaken* and *generalize* premise and conclusion.

Paths and accessibility

Let $\prec$ be an arbitrary binary relation.

$$\begin{aligned} \mathbf{Path}_\prec(x) &\overset{\nu}{=} \exists y \prec x\, \mathbf{Path}_\prec(y) \qquad (\nu \text{ means 'greatest'}) \\ \mathbf{Acc}_\prec(x) &\overset{\mu}{=} \forall y \prec x\, \mathbf{Acc}_\prec(y) \end{aligned}$$

Classically, $\mathbf{Path}_\prec$ and $\mathbf{Acc}_\prec$ are complements of each other.

$\mathbf{Path}_\prec(x)$ means (with dependent choice) that there is an infinite $\prec$-descending sequence starting with x.

$\mathbf{Acc}_\prec(x)$ means that $\prec$-induction is valid at x.

Setting $s \prec_P t \overset{\text{Def}}{=} \exists n\, s = t * n \wedge \neg P(t)$:

$\neg\mathbf{Path}_{\prec_P}(\langle\rangle)$ means that P is a bar,

$\mathbf{Acc}_{\prec_P}(\langle\rangle)$ means that P is an inductive bar, provided P is decidable.

Brouwer's thesis without computational content

The implication $\mathbf{Acc}_{\prec}(x) \to \neg\mathbf{Path}_{\prec}(x)$ is intuitionistically valid (easy $\prec$-induction).

The converse is can be viewed as a version of Brouwer's thesis:

$\mathbf{BT}_0$ $\forall x\,(\neg\mathbf{Path}_{\prec}(x) \to \mathbf{Acc}_{\prec}(x))$

Both, the premise and conclusion of $\mathbf{BT}_0$, are Harrop formulas (do not contain $\vee$ or $\exists$ at a strictly positive position).

Therefore, $\mathbf{BT}_0$ has no computational content and hence does not spoil program extraction.

Abstract bar induction (**ABI**)

$$y \prec^* x \quad \stackrel{\mu}{=} \quad y = x \vee \exists z\,(y \prec^* z \wedge z \prec x) \qquad \text{(refl. trans. closure)}$$
$$y \prec_P x \quad \stackrel{\text{Def}}{=} \quad y \prec x \wedge \neg P(x)$$

Let x_0 be arbitrary (playing the role of the empty sequence).

ABI If
(1) $\neg\mathbf{Path}_{\prec_P}(x_0)$
(2) $\forall x \prec^* x_0\,(\neg P(x) \vee Q(x))$,
(3) $\forall x \prec^* x_0\,(\forall y \prec x\, Q(y) \to Q(x))$,
then $Q(x_0)$.

Lemma. $\mathbf{BT}_0$ implies **ABI**.

Proof. Assume (1), (2), (3). By $\mathbf{BT}_0$, $\mathbf{Acc}_{\prec_P}(x_0)$. We prove $\mathbf{Acc}_{\prec_P} \subseteq Q$ by wellfounded induction. By i.h., $\forall y \prec_P^* x\, Q(y)$. We have to show $Q(x)$. We do a case analysis according to (2). If $Q(x)$, we are done. If $\neg P(x)$ then the i.h. is equivalent to the premise of (3), hence, again $Q(x)$.

Vacuous truth

If A is a formula, then $\mathcal{V}(A)$ is a Harrop formula with $\_\,\mathbf{r}\,\mathcal{V}(A) \stackrel{\text{Def}}{=} \forall a\,(a\,\mathbf{r}\,A)$.

For example, $\_\,\mathbf{r}\,\mathcal{V}(\bot \to A)$ since, $a\,\mathbf{r}\,(\bot \to A) \equiv \bot \to a\,\mathbf{r}\,A$.

Intuitively, $\mathcal{V}(A)$ expresses that A is vacuously true or true (realizable) for trivial reasons.

Valid (realizable) rules we will use in the following:

$$\frac{A}{\mathcal{V}(A)} \qquad (A \text{ Harrop})$$

$$\frac{A \to \mathcal{V}(B)}{\mathcal{V}(A \to B)} \qquad \frac{\mathcal{V}(A) \wedge \mathcal{V}(B)}{\mathcal{V}(B \wedge A)}$$

$$\frac{\forall x\,\mathcal{V}(A(x))}{\mathcal{V}(\forall x\,A(x))} \qquad \frac{\exists x\,\mathcal{V}(A(x))}{\mathcal{V}(\exists x\,A(x))}$$

$\mathbf{LEM}_{\mathcal{V}}$, a realizable law of excluded middle

$$\frac{\neg A \to B \qquad A \to \mathcal{V}(B)}{B}$$

Lemma (LEM)

The rules for $\mathcal{V}(\cdot)$ are realizable.

Proof.

We only look at $\mathbf{LEM}_{\mathcal{V}}$.

Assume $a\,\mathbf{r}\,(\neg A \to B)$ and $\_\,\mathbf{r}\,(A \to \mathcal{V}(B))$, that is, $(\neg\exists c\; c\,\mathbf{r}\,A) \to a\,\mathbf{r}\,B$ and $(\exists c\; c\,\mathbf{r}\,A) \to \forall b\; b\,\mathbf{r}\,B$.

Using the law of excluded middle, we conclude $a\,\mathbf{r}\,B$. □

Abstract bar induction with vacuous bars

ABI$_\mathcal{V}$ If

(1) $\neg\mathbf{Path}_{\prec_P}(x_0)$,
(2) $\forall x \prec^* x_0\,(P(x) \to \mathcal{V}(Q(x)))$,
(3) $\forall x \prec^* x_0\,(\forall y \prec x\, Q(y) \to Q(x))$,

then $Q(x_0)$.

Lemma

BT$_0$ *implies* **ABI**$_\mathcal{V}$.

Proof.

The proof is almost identical to the proof for **ABI**. The only difference is that we use **LEM**$_\mathcal{V}$ to do a case analysis, on whether $P(x)$ holds, using (2). □

The extracted program takes as input a realizer g of (3) (note that (2) is Harrop) and returns $h\,\langle\rangle$ where

$$h\,s = g\,s\,(\lambda a\,(h\,(s * a))).$$

Proving uniform continuity

We aim to prove that every total continuous functional F on Cantor space is uniformly continuous and extract from the proof the *fan functional* that computes the minimal modulus of uniform continuity of F.

Language:

Sorts: s_0 (partial natural numbers), s_1 ($\simeq s_0 \to s_0$), s_2 ($\simeq s_1 \to s_0$).
Constants: $0, 1, \bot$, where $0, 1$ represent at the same time the first two natural numbers and the Booleans, and $\bot$ represents 'undefined' (not to be confused with the formula $\bot$).
Function symbols: $+, -$, application operation (written by juxtaposition), common (primitive recursive) operations to define finite and infinite sequences.
Relation symbol: $<$ (ordinary ordering of numbers).

Axioms: The usual disjunctions-free axioms for $0, 1, +, -, <$.

Natural numbers: $\mathbf{N}(x) \stackrel{\mu}{=} x = 0 \vee \mathbf{N}(x - 1)$.

Booleans: $\mathbb{B}(x) \stackrel{\text{Def}}{=} x = 0 \vee x = 1$

Partial functionals

We define the partial Booleans and natural numbers as well as the partial functionals of type 1 and 2:

$$
\begin{aligned}
\mathbb{B}_\perp(x) &\stackrel{\text{Def}}{=} x \neq \perp \to \mathbb{B}(x)\\
\mathbf{N}_\perp(x) &\stackrel{\text{Def}}{=} x \neq \perp \to \mathbf{N}(x)\\
\mathbb{B}^1_\perp(\alpha) &\stackrel{\text{Def}}{=} \forall n\,(\mathbf{N}(n) \to \mathbb{B}_\perp(\alpha\, n))\\
\mathbb{B}^2_\perp(F) &\stackrel{\text{Def}}{=} \forall \alpha\,(\mathbb{B}^1_\perp(\alpha) \to \mathbf{N}_\perp(F\alpha))
\end{aligned}
$$

Continuity

Specialization order:

$$\begin{aligned} x \sqsubseteq y &\stackrel{\text{Def}}{=} x \neq \bot \to x = y \\ \alpha \sqsubseteq \beta &\stackrel{\text{Def}}{=} \forall n \in \mathbf{N}\,(\alpha\, n \sqsubseteq \beta\, n) \end{aligned}$$

Monotonicity, *finitarity*, *continuity*:

$$\begin{aligned} \mathbf{Mon}(F) &\stackrel{\text{Def}}{=} \forall \alpha, \beta \in \mathbb{B}^1_\bot\,(\alpha \sqsubseteq \beta \to F\,\alpha \sqsubseteq F\,\beta) \\ \mathbf{Fin}(F) &\stackrel{\text{Def}}{=} \forall \alpha \in \mathbb{B}^1_\bot\,(\forall n \in \mathbf{N}\, F\,(\alpha \upharpoonright n) = \bot \to F\,\alpha = \bot) \\ \mathbf{Cont}(F) &\stackrel{\text{Def}}{=} \mathbf{Mon}(F) \wedge \mathbf{Fin}(F) \end{aligned}$$

where $(\alpha \upharpoonright n)\, k = \mathbf{if}\ k < n\ \mathbf{then}\ \alpha\, k\ \mathbf{else}\ \bot$.

Totality

$$\begin{aligned}
\mathbf{Total}^1(\alpha) &\stackrel{\text{Def}}{=} \forall n\,(\mathbf{N}(n) \to \alpha\, n \neq \bot) \\
\mathbf{Total}^2(F) &\stackrel{\text{Def}}{=} \forall \alpha\,(\mathbf{Total}^1(\alpha) \to F\alpha \neq \bot))
\end{aligned}$$

$$\begin{aligned}
\mathbb{B}^1(\alpha) &\stackrel{\text{Def}}{=} \mathbb{B}^1_\bot(\alpha) \wedge \mathbf{Total}^1(\alpha) \\
\mathbb{B}^2(F) &\stackrel{\text{Def}}{=} \mathbb{B}^2_\bot(F) \wedge \mathbf{Total}^1(F)
\end{aligned}$$

Uniform continuity

A type 2 functional F is *uniformly continuous* if there is (a least) $n \in \mathbf{N}$ such that $F\,\alpha = F\,\beta$ for all total α, β agreeing below n.

$$\begin{aligned} \mathbf{UCont}(F, n) &\stackrel{\text{Def}}{=} \forall \alpha, \beta \in \mathbb{B}^1(\alpha =_n \beta \to F\,\alpha = F\,\beta) \\ \mathbf{UCont}(F) &\stackrel{\text{Def}}{=} \exists n \in \mathbf{N}\, \mathbf{UCont}(F, n) \end{aligned}$$

where $\alpha =_n \beta \stackrel{\text{Def}}{=} \forall k \in \mathbf{N}\,(k < n \to \alpha\, k = \beta\, k)$.

We aim to prove that every $F \in \mathbb{B}^2_\perp$ which is total and continuous is uniformly continuous.

Extremal points

In the following let F be a total continuous functional, that is, $F \in \mathbb{B}^2$ and $\mathbf{Cont}(F)$.

Theorem (Existence of extremal points)

$$\exists \alpha_{\mathsf{min}}, \alpha_{max} \in \mathbb{B}^1 \, \forall \beta \in \mathbb{B}^1 \, (F(\alpha_{\mathsf{min}}) \leq F(\beta) \leq F(\alpha_{\mathsf{max}}))$$

Proof.

Let $\mathbb{B}^*$ be the set of finite sequences of Booleans, that is,

$$\mathbb{B}^*(s) \stackrel{\mu}{=} s = \langle\rangle \vee \exists t \in \mathbb{B}^* \, \exists b \in \mathbb{B} \, s = t * b,$$

Define $(s * \alpha)\, n = s_n$ if $n < |s|$ and $(s * \alpha)\, n = \alpha(n - |s|)$ if $n \geq |s|$.

We prove more generally (max only, for min we proceed similarly):

$$\forall s \in \mathbb{B}^* \exists \alpha_{max} \in \mathbb{B}^1 \, \forall \beta \in \mathbb{B}^1 \, (F(s * \beta) \leq F(s * \alpha_{\mathsf{max}}))$$

Proof of the existence of extremal points ctd.

$$\begin{aligned} \mathbf{sec}(s) &\stackrel{\text{Def}}{=} F\,(s * \bot^{\omega}) \neq \bot \quad (\text{'}s \text{ is secured'}) \\ s \prec t &\stackrel{\text{Def}}{=} \exists b \in \mathbb{B}\, s = t * b \end{aligned}$$

Hence $\mathbb{B}^*(s)$ iff $s \prec^* \langle\rangle$.

$$\begin{aligned} Q(s,\alpha) &\stackrel{\text{Def}}{=} F(s*\alpha) \neq \bot \wedge \\ &\qquad \forall \beta \in \mathbb{B}^1_\bot \,(F(s*\beta) \neq \bot \to F(s*\beta) \leq F(s*\alpha)) \\ Q(s) &\stackrel{\text{Def}}{=} \exists \alpha \in \mathbb{B}^1_\bot \, Q(s,\alpha) \end{aligned}$$

Claim: $\forall s \in \mathbb{B}^*\, Q(s)$.

We prove the claim by $\mathbf{ABI}_{\mathcal{V}}$ on $\prec_{\mathbf{sec}}$.

Applying $\mathbf{ABI}_{\mathcal{V}}$

We have to show

(1) $\forall s \in \mathbb{B}^* \neg \mathbf{Path}_{\prec_{\mathbf{sec}}}(s)$,

(2) $\forall s \in \mathbb{B}^* (\mathbf{sec}(s) \to \mathcal{V}(Q(s)))$,

(3) $\forall s \in \mathbb{B}^* (\forall a \in \mathbb{B}\, Q(s * a) \to Q(s))$,

(1) holds F since is total and continuous.

(2): By efq, $\to_{\mathcal{V}}$, and $\forall_{\mathcal{V}}$, $\mathcal{V}(\mathbb{B}^1_\perp(\perp^\omega))$. If $s \in \mathbb{B}^*$ is secured, then clearly $Q(s, \perp^\omega)$. Since this a Harrop formula, it follows $\mathcal{V}(Q(s, \perp^\omega))$, by $\mathbf{H}_{\mathcal{V}}$. With $\wedge_{\mathcal{V}}$ and $\exists_{\mathcal{V}}$ it follows $\mathcal{V}(Q(s))$.

(3): Let $s \in \mathbb{B}^*$ such that $\forall a \in \mathbb{B}\, Q(s * a)$, that is, we have $\alpha_0, \alpha_1 \in \mathbb{B}^1_\perp$ such that $Q(s * 0, \alpha_0)$ and $Q(s * 1, \alpha_1)$. We have to find $\alpha \in \mathbb{B}^1_\perp$ such that $Q(s, \alpha)$. Since $F \in \mathbb{B}^2_\perp$, we have $F(s * 0 * \alpha_0), F(s * 1 * \alpha_1) \in \mathbf{N}$. If $F(s * 0 * \alpha_0) \geq F(s * 1 * \alpha_1)$, set $\alpha = 0 * \alpha_0$. Otherwise, set $\alpha = 1 * \alpha_1$.

This completes the proof of the Claim and hence the Theorem.

Deciding constancy

$$\mathbf{Const}(F, s) \stackrel{\text{Def}}{=} \exists b \in \mathbb{B}\, \forall \alpha \in \mathbb{B}^1\, F\,(s * \alpha) = b$$

Theorem (Decidability of constancy)

For every $s \in \mathbb{B}^*$ it is decidable whether F is constant on total extensions of s, that is, $\mathbf{Const}(F, s) \vee \neg\mathbf{Const}(F, s)$.

Proof.

By the theorem about the existence of extremal points there are $\alpha_{min} \in \mathbb{B}^1\, \alpha_{max} \in \mathbb{B}^1$ such that $F(s * \alpha_{\mathsf{min}}), F(s * \alpha_{\mathsf{max}}) \in \mathbf{N}$ and for all $\beta \in \mathbb{B}^1$

$$F(s * \alpha_{\mathsf{min}}) \leq F(s * \beta) \leq F(s * \alpha_{\mathsf{max}})$$

Hence $\mathbf{Const}(F, s)$ holds iff $F(s * \alpha_{\mathsf{min}}) = F(s * \alpha_{\mathsf{max}})$. Since equality of natural numbers is decidable the theorem follows.

The proof of uniform continuity

Theorem

Every functional $F \in \mathbb{B}^2_\perp$ which is total and continuous is uniformly continuous.

Proof.

Let $F \in \mathbb{B}^2_\perp$ be total and continuous. We set

$$\begin{aligned} \mathbf{UCont}(s,n) &\stackrel{\text{Def}}{=} \forall \alpha,\beta \in \mathbb{B}^1(\alpha =_n \beta \to F(s*\alpha) = F(s*\beta)) \\ \mathbf{UCont}(s) &\stackrel{\text{Def}}{=} \exists n \in \mathbf{N}\, \mathbf{UCont}(s,n) \end{aligned}$$

and show $\forall s \in \mathbb{B}^*\, \mathbf{UCont}(s)$ by abstract bar induction, **ABI**, on $\prec_{\mathbf{Const}}$ where $\prec$ is as in the proof of the existence of extremal points $\mathbf{Const}(s) \stackrel{\text{Def}}{=} \mathbf{Const}(F,s)$.

Applying **ABI**

We have to show:

(1) $\mathbf{Wf}_{\prec_{\mathbf{Const}}}(\langle\rangle)$,

(2) $\forall s \in \mathbb{B}^* \, (\neg \mathbf{Const}(s) \vee \mathbf{UCont}(s))$,

(3) $\forall s \in \mathbb{B}^* \, (\forall a \in \mathbb{B} \, \mathbf{UCont}(s * a) \to \mathbf{UCont}(s))$.

(1) holds again by continuity.

(2): By the Constancy Theorem, we may assume $\mathbf{Const}(s)$. Then clearly $\mathbf{UCont}(s, 0)$.

(3): Assume $\mathbf{UCont}(s * 0, n)$ and $\mathbf{UCont}(s * 1, m)$. Then, clearly, $\mathbf{UCont}(s, 1 + \max(n, m))$.

Program extraction

Declarations:

```
type N = Int
type B = Int
type B1 = N -> B
type B2 = B1 -> N

(***) :: [B] -> B1 -> B1
s *** alpha = \n-> if n < length s
                  then s !! n
                  else alpha (n - length s)
```

Computing extremal points

```
minarg, maxarg :: B2 -> [B] -> B1

minarg f s = let {
                   s0 = s ++ [0] ; s1 = s ++ [1] ;
                   alpha0 = minarg f s0 ;
                   alpha1 = minarg f s1
                 }
             in if f (s0 *** alpha0) <= f (s1 *** alpha1)
                then [0] *** alpha0
                else [1] *** alpha1

maxarg f s = ...
```

Fan functional

```
-- testing constancy
isconst :: B2 -> [B] -> Bool
isconst f s =
   f (s *** (minarg f s)) == f (s *** (maxarg f s))

fan :: B2 -> N
fan f = aux []

  where

--  aux :: [B] -> N
    aux s = if isconst f s
            then 0
            else 1 + max (aux (s++[0])) (aux (s++[1]))
```

The origin of this program is unclear. Martin Hyland claims it was known already to Robin Gandy.

Further work in program extraction

Realizability and program extraction is implemented in the interactive proof system *Minlog* developed by H Schwichtenberg in Munich.

`http://www.mathematik.uni-muenchen.de/~logik/minlog/`

Overview of existing case studies in program extraction

- Discrete structures
  - Quotient and remainder on natural numbers.
  - Dijkstra's algorithm (1997, Benl, Schwichtenberg):
    Reachable nodes in a weighted graph
  - Warshall Algorithm (2001, Schwichtenberg, Seisenberger, B):
    Transitive closure of a relation
- Programs from classical proofs
  - GCD (1995, B, Schwichtenberg):
    Uses the Friedman/Dragalin A-translation
  - Dickson's Lemma (2001, Schwichtenberg, Seisenberger, B):
    F/D A-translation in infinite combinatorics
  - Higman's Lemma (2008, Seisenberger):
    Uses F/D A-translation and classical countable choice
  - Fibonacci numbers from a classical proofs (2002, Buchholz, Schwichtenberg, B):
    Uses F/D A-translation to obtain fast program

Overview ctd.

- Lambda calculus:
  - Extraction of normalization-by-evaluation (NbE) (2006, Berghofer, Letouzey, Schwichtenberg, B):
    Extraction of NbE from Tait's proof of strong normalization for the typed lambda calculus (in Isabelle, Coq, Minlog)

- Real numbers
  - Cauchy sequences vs signed digit representation (SD):
    Cauchy sequences are functions.
    SD representations are streams defined by coinduction.
  - Arithmetic operations on reals w.r.t. SD
  - Integration w.r.t. SD (2011, B):
    Real functions are given by trees realizing a nested coinductive/inductive definition

Overview ctd.

- Lists
  - List reversal
    Uses F/D A-translation to extract linear program from naive proof
  - In-place Quicksort (2014, Seisenberger, Woods, B):
    Extracts an 'imperative' program

- Satisfiabilty testing
  - Extraction of a SAT-solver from completeness proof for DPLL (2015, B, Forsberg, Lawrence, Seisenberger)

- Ongoing: Extraction of
  - monadic parsers (Jones, Seisenberger, B)
  - concurrent programs (Miyamoto, Petrovska, Schwichtenberg, Spreen, Takayama, Tsuiki, B)
  - truly imperative programs (Reus, B)
  - modulus of uniform continuity from Fan Theorem (B)

Conclusion

- The fine grained control of computational content not only optimizes extracted programs but also provides access to new kinds of algorithms by program extraction.
- Limited use of classical logic seems to be required to verify the correctness of these new algorithms.
- The Harrop version of Brouwer's thesis and bar induction with vacuous bar might open ways to extract programs such as the Berard-Bezem-Coquand realizer of dependent choice from a proof.

References

Hideki Tsuiki. Real Number Computation through Gray Code Embedding.
Theor. Comput. Sci., *284(2):467–485, 2002.*

B., Kenji Miyamoto, Helmut Schwichtenberg, Hideki Tsuiki: Logic for Gray-code computation. In: Concepts of Proof in Mathematics, Philosophy, and Computer Science, de Gruyter, 2016.

B., Extracting Non-Deterministic Concurrent Programs. CSL 2016, LIPICS

L. E. J. Brouwer, Beweis dass jede volle Funktion gleichmässig stetig ist.
Nederlandse Akademie van Wetenschappen Verslagen 27, 189193, 1924.

L. E. J. Brouwer, Über Definitionsbereiche von Funktionen,
Math. Annalen 97, 6075, 1927.

References

V. Veldman. Brouwer's Real Thesis on Bars.
Philosophia Scientiæ, CS 6, *Constructivism: Mathematics, Logic, 21-42 Philosophy and Linguistics, 2006.*

H. Schwichtenberg. Minlog.
The Seventeen Provers of the World, Lecture Notes in Artificial Intell., *3600, 151–157, 2006.*
`http://www.mathematik.uni-muenchen.de/~logik/minlog/`

M. Escardó. Exhaustible sets in higher-type computation,
Logical Methods in Comput. Sci. *4 (3), 2008.*

B. Totale Objekte und Mengen in der Bereichstheorie,
PhD thesis, LMU Munich, *1990.*

References

B. From coinductive proofs to exact real arithmetic: theory and applications.
Logical Methods in Comput. Sci., *7(1):1–24, 2011.*

M. Escardo, P. Oliva. Bar recursion and products of selection functions, *JSL, 80(1):1-28, 2015.*

B, O. Petrovska Optimized program extraction for induction and coinduction
CiE 2018, LNCS 10936, 70-80, 2018.